THE MARCH 11TH SESSION

- Energy in water utilities (20 min)
- Introduction to pumping and pump design (45 min)
- Pump modeling exercise (30 min)
- Break (10 min)
- Introduction to optimization (25 min)
- Optimization in water sector (25 min)
- Modeling exercise – energy, leakage, pumping (30 min)

OPTIMIZATION OF WSS

Aalto University – 2019-03-11

EUR ING PhD MARKUS SUNELA

LEARNING OUTCOMES

- List three use cases for optimization in WSS
- List commonly used constraints and their values in WSS optimization
- Describe the complexities associated with optimization of WSS
- Describe how genetic algorithm works

THREE SIDES OF WSS OPTIMIZATION

- Design optimization
  - Part of everything the engineers normally do manually: assessing different options and choosing the best
  - Planning and optimizing new areas, pipes, rehabilitation projects, tanks, stations, pumps, control systems, water resources management etc.

- Operational optimization
  - Offline optimization
    - Best possible solutions are found out using optimization algorithms or other methods under a limited number of scenarios before hand, relative volumes pumped from different sources, pumping settings, preparing for exceptions, pipe bursts
  - Online optimization
    - Optimization program calculates the best solution dynamically and continuously: for example the settings for all presssure booster stations or water sources
    - Can be controlling or an expert system

- Calibration
  - Change model parameters so that the simulated results match the measured
  - Locating leaks, closed valves, general hydraulic or quality calibration

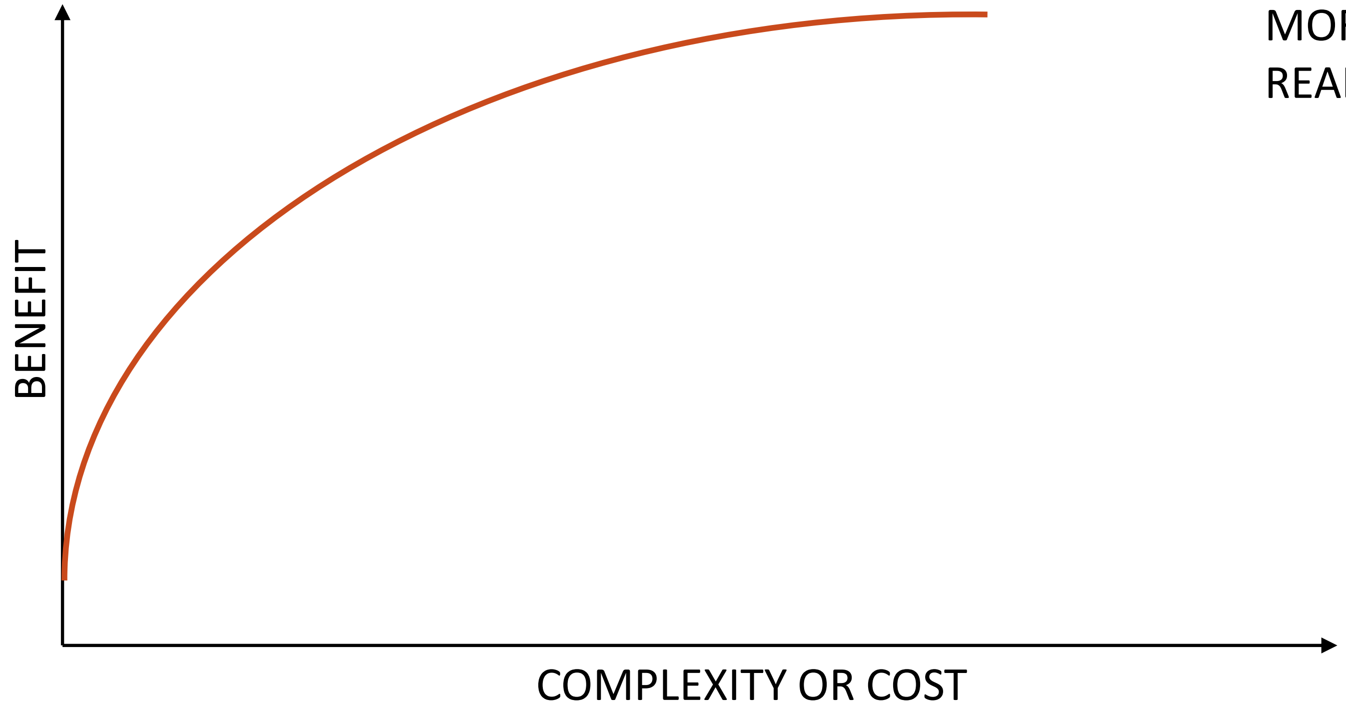
DIFFERENT LEVELS OF ACTIONS TO REDUCE ENERGY USE

- Changing control system parameters and settings
  - Constant, smooth pumping, using of multiple directions at once, utilizing the sources close to demand maximally
  - Fully automatic control
  - Optimizing parallel pumping, processes

- Replacing pumps
  - Installing variable speed drives
  - Installing new pumps, possibly multiple differently sized pumps

- Changes in the network
  - Pressure re-zoning
  - Adding more elevated storage
  - Restructuring storage: lowering or rising the tanks
  - Fixing, constructing or redimensioning of pipes

COST VS BENEFIT



AFTER SOME POINT PUTTING MORE EFFORT DOESN'T REALLY PAY BACK ANYMORE

TYPICAL CONSTRAINTS

- Users
  - Pressure level (2–8 bar)
  - Pressure difference (<1 bar)
  - Quality, especially water age (<4–7 days)
- Network
  - Velocity (<0.5–1.0 m/s)
  - Unit headloss (<3–5‰)
- Tanks
  - Minima and maxima for level, volume and capacity (hours)
  - Quality, especially water age (<4–7 days)

TYPICAL CONSTRAINTS

- Water soures
  - Minimum and maximum flow (hourly and daily constraints)
  - Minimum running time for pumps and maximum allowed number of pump switches
  - Outlet pressure limit (8–10 bar)
  - Pump capacity
  - Limits for efficiency, power and specific energy

- Pressure booster stations
  - Inlet pressure limit (1 bar)
  - Outlet pressure limit (8–10 bar)
  - Pump capacity
  - Limits for efficiency, power and specific energy

WHY OPTIMIZATION IS HARD IN WSS?

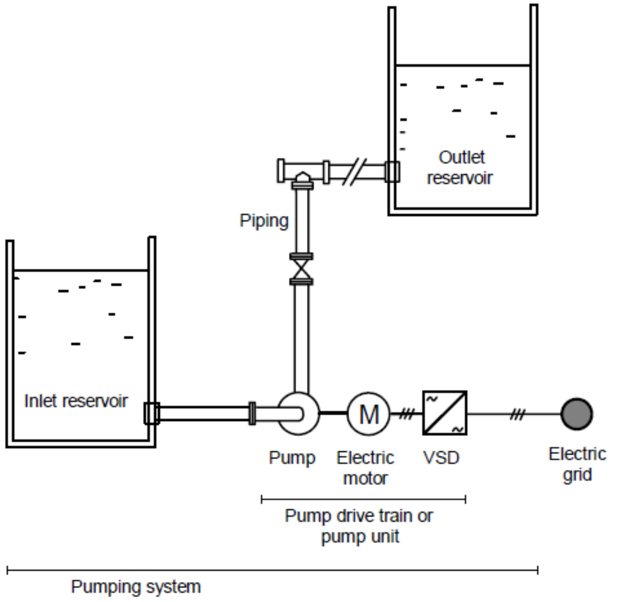
- In pressurized system everything affects everything else
  - Even the smallest system has numerous solutions
  - Change in controls, outlet pressure or tank volume can cause surprizing cascading effects

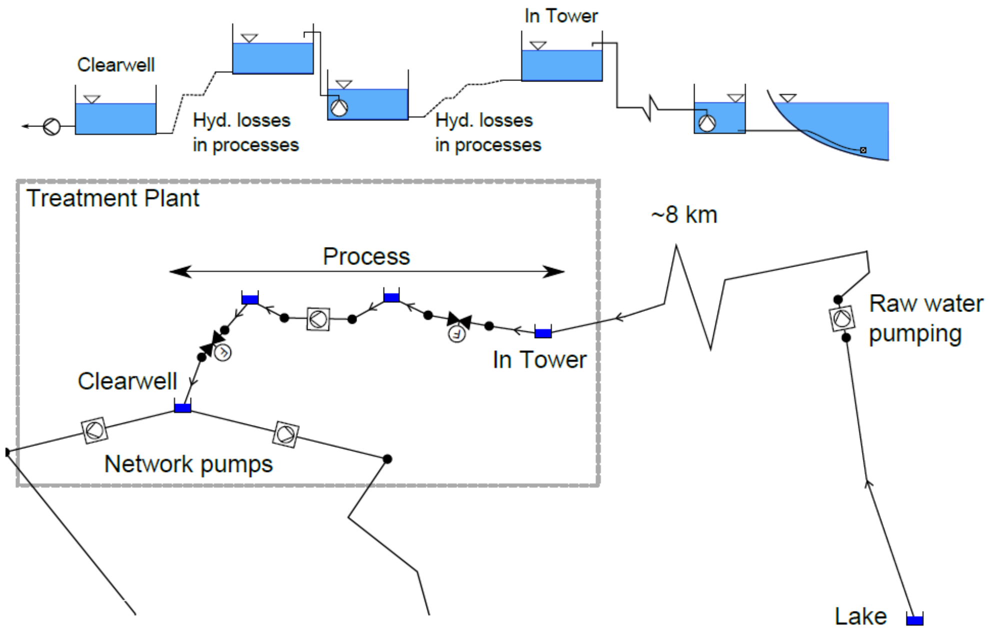
- Network equations cannot be solved analytically and they are non-linear
  - Traditional methods cannot be used
  - Optimization is computationally intensive

- Evaluating constraints, and often the objective function too, require hydraulic modeling

HYDRAULIC MODEL

- Optimization requires the use of a hydraulic model (EPANET, EPASWMM)
- Model solves the energy use and workings of the network for the given solution candidate: objective + constraints
- Model accuracy should reflect the desired goal: typically as accurate as possible, but computational time can limit accuracy
  - All pipes, every water user, leakage
  - All variable speed drives, motors, pumps, control algorithms





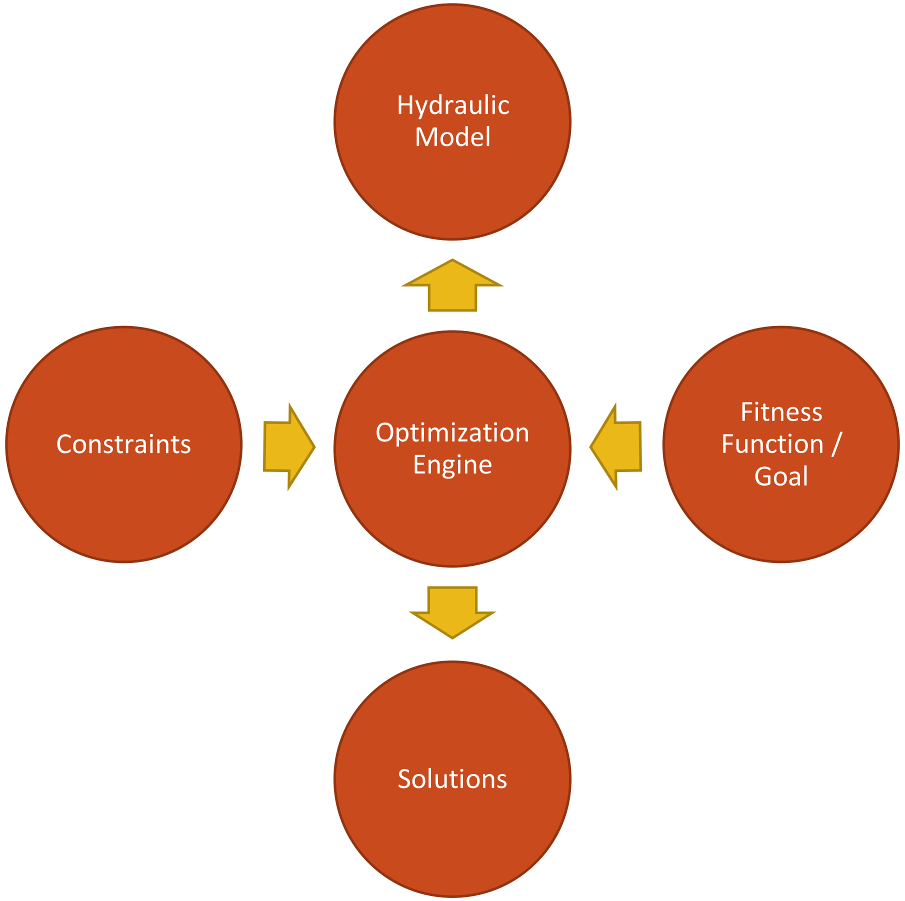
EPANET

- Public domain, basis for practically all commercial solutions: WaterCAD, Fluidit Water, MikeUrban…
- Under active scientific research
- Has its limitations tough
  - Parallel pumping and variable speed control
  - Inaccuracies in efficiency calculations
  - Limited possibilities for controlling the network
  - ~~Not thread-safe nor re-entrant~~ (implemented in up-coming 2.2)

- A lot of fixes and feature present in literature
  - Sunela 2015b, 2015c, 2016, 2017
  - Marchi & Simpson 2013

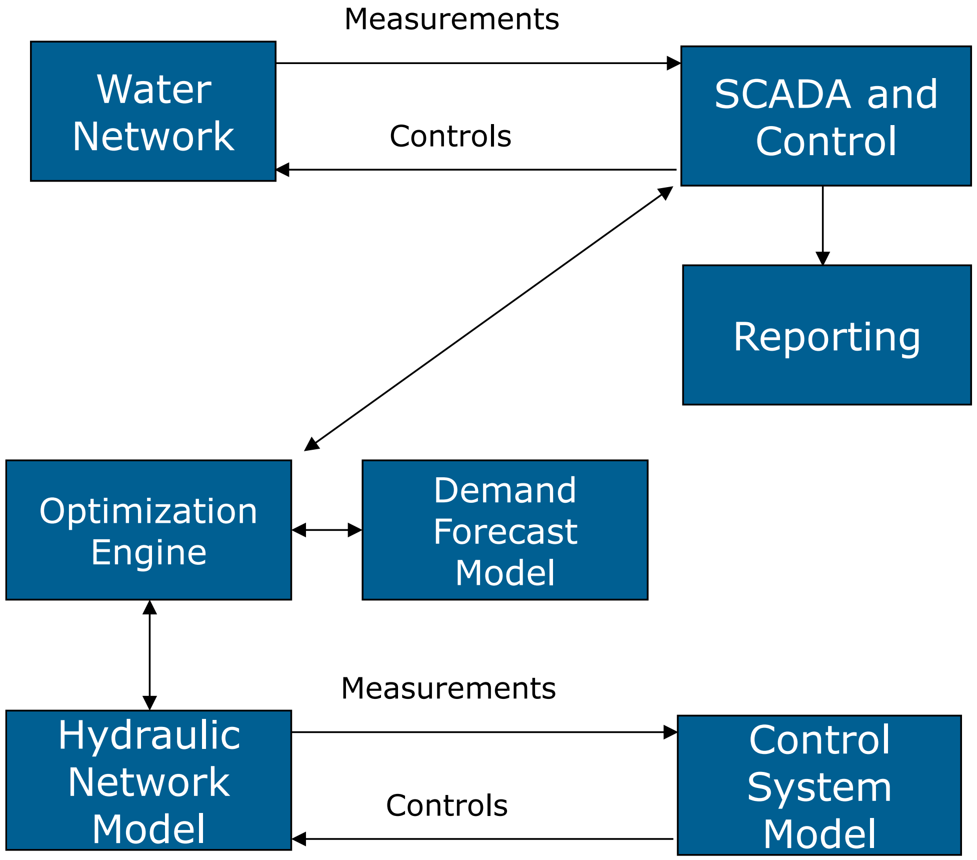
- Easy to use in own code (C, Java, Python, C#...) and in, for example, Matlab and Excel
- Some more recent developments available through OpenWaterAnalytics project

HOW TO OPTIMIZE?

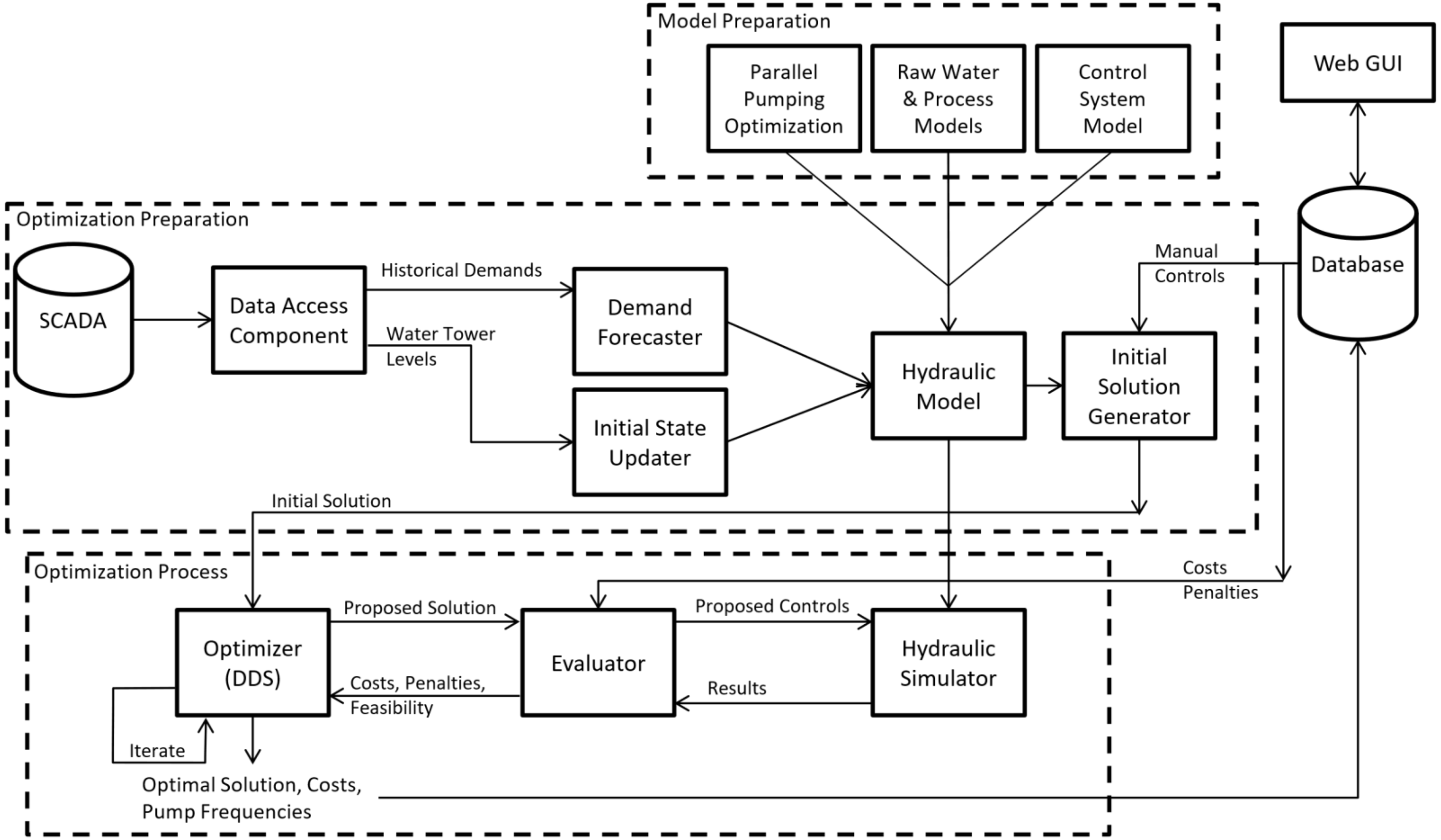


ONLINE OPTIMIZATION

- Requires two-way connection between SCADA and optimization tool
- Typically optimization is done once an hour, for the next 24 h
  - Initial tank levels
  - Demand forecast
  - Optimal settings for every optimizable station for each hour
  - Constraint definitions
  - Errors and problems with data



EXAMPLE IMPLEMENTATION



Sunela 2017

OPTIMIZATION METHODS FOR WSS

- WSS optimization is NP hard problem – **only approximate solutions exist**
- Traditional optimization methods (LP, NLP, DP…) work poorly, if at all, and require a lot of time to formulate the problem properly

- Meta-heuristic algorithms are commonly used
  - Trajectory based vs population based
  - **No guarantee of finding the global optimum**, but results are ”good enough”
  - **Don’t require analytical solutions or derivatives**. Instead the system is treated as black box, that only returns the objective function value and feasibility (for example using a hydraulic model). The solutions are made better iteratively using heuristic methods.
  - **Require a lot of computational power**

COMPUTATIONAL TIME

- Simulations are relatively slow, and the number of simulations is great
- The significance of computational time is even more apparent in online applications
- Can be improved using
  - Parallel processing – many candidates at once or parallelized simulator
  - Hybrid algorithms
  - Caching
  - Model simplification (surrogate model)
  - Multi-level optimization or multi-level evaluation (problem decomposition)
  - Probabilistic model building GA (PMBGA)
  - Combination of the above
  - Clever problem formulation to avoid **The Curse of Dimensionality**: grouping, decomposition, restricting design variable value ranges, solving deltas/fractions…

SOME META-HEURISTIC ALGORITHMS

- Evolutionary algorithms (EA)
  - Model evolution of biological populations
  - Genetic algorithm (GA) is the most commonly used variant
- Swarm algorithms (SA, swarm intelligence)
  - Model movement and behavior of insect and animal swarms and colonies
  - Most commonly used are particle swarm optimization (PSO) and ant colony optimization (ACO)
- Others, such as dynamically dimensioned search (DDS)

PARTICLE SWARM ALGORITHM

- Is modeled after swarms of birds and fishes
- Population, or swarm, of candidate solutions or particles
  - Initially particles are distributed evenly throughout the search base
- Every particle has a location and velocity
  - Coordinate system is n dimensional, where n is the number of design variables
  - Each coordinate is value of a design variable
- The velocity for each particle is updated every iteration
  - According to simple equations
  - The change depends on the particle's own best result and the best result of the whole swarm
  - Location updated based on the velocity
- Basic implementation gets easily stuck in local optimum
- A lot of different versions exist

ANT COLONY OPTIMIZATION

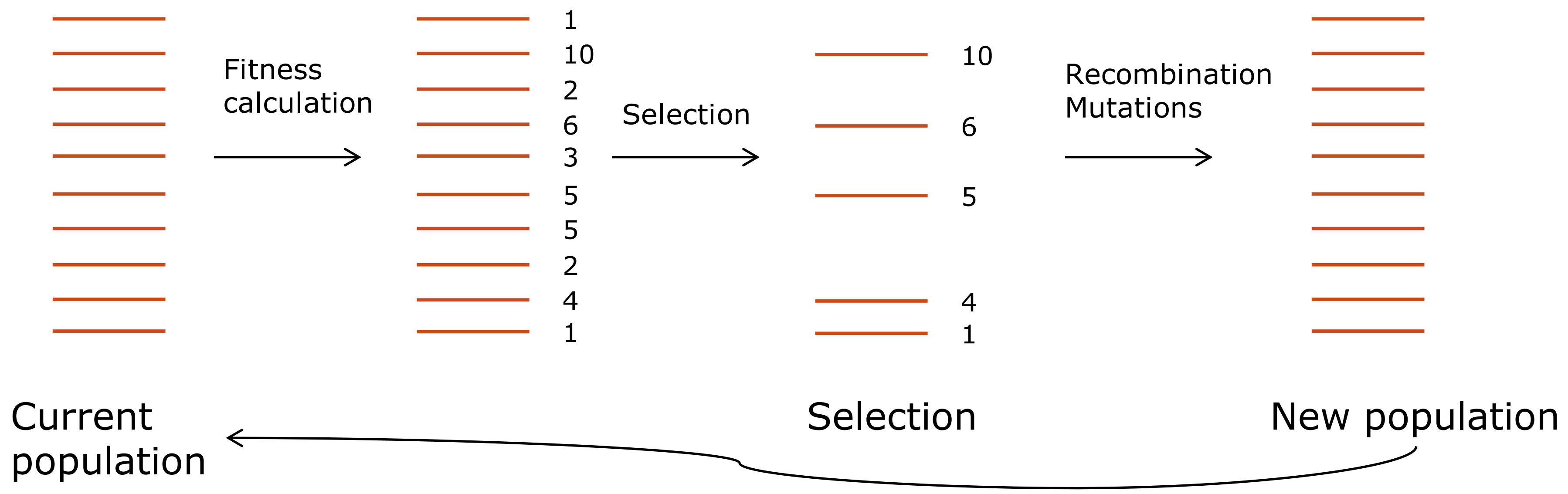
- Modeled after the way how ants optimize the route between food source and colony
- Initially the ants walk randomly
- When food is found, they return to the colony and leave a trail of pheromones
- The next iteration ants are more likely to choose a route with more pheromons
- The more ants choose an route, the more pheromones are excreted and the more likely it is that other ant choose the route
- The amount of pheromones an ant excretes is depends on the goodness of the solution
- Pheromones evaporate partly every iteration
- Works only with integer valued design variables: every variables in a road junction, where the range of valid values are the different routes continuing from the junction

GENETIC ALGORITHM

- Is modeled after evolution – how DNA is evolved
  - Population of chromosomes (solution candidate) – each chromosome has multiple genes (design variables)
  - The fitness of each solution is calculated every iteration
  - The best or most fit solutions are most likely to reproduce and have descendants in the next generation
  - The chromosomes exchange genes and mutate during the process

- Encoding design variables (genes) into chromosomes
  - Most typical solution is using a bit string
    - One gene is formed by 1–n bit
    - Bit coding is stored in a table (for example pipe diameter coding: 00=63 mm, 01=110 mm, 10=160 mm and 11=225 mm)
  - Integer and real valued genes are possible too, but bitstrings are the most researched

GENETIC ALGORITHM



GENETIC ALGORITHM PARAMETERS

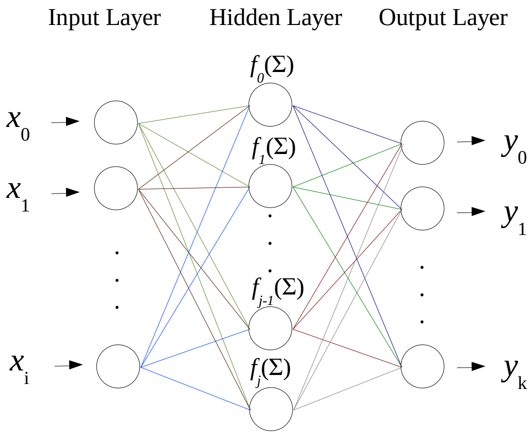
- Parameters
  - Population size, typically 5–10 times the number of design variables
  - Probability of recombination 0.5–0.7
  - Probability of mutation typically very small 0.001

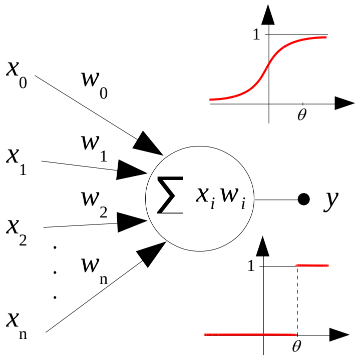
- Some of the chromosomes continue unchanged from one generation to the next
- Usually 10 to 20 best solutions found so far are saved unchanged in the population

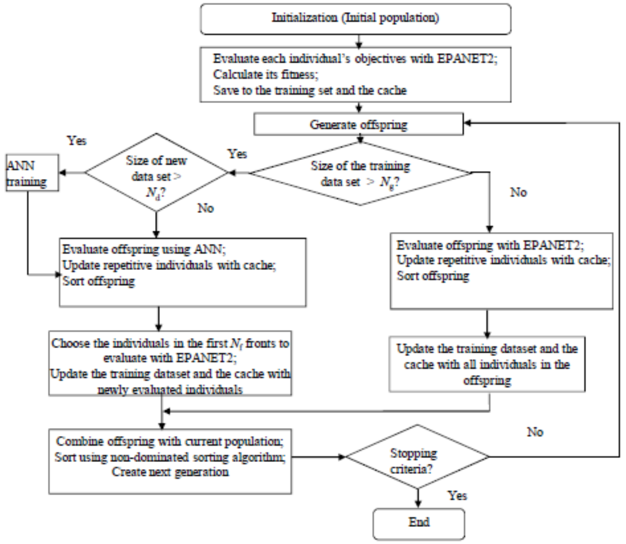
- Depending on the problem, there can be from tens to hundreds of thousands of generation and each generation requires a number of simulations equal to the number of population size

ARTIFICIAL NEURAL NETWORK (ANN)

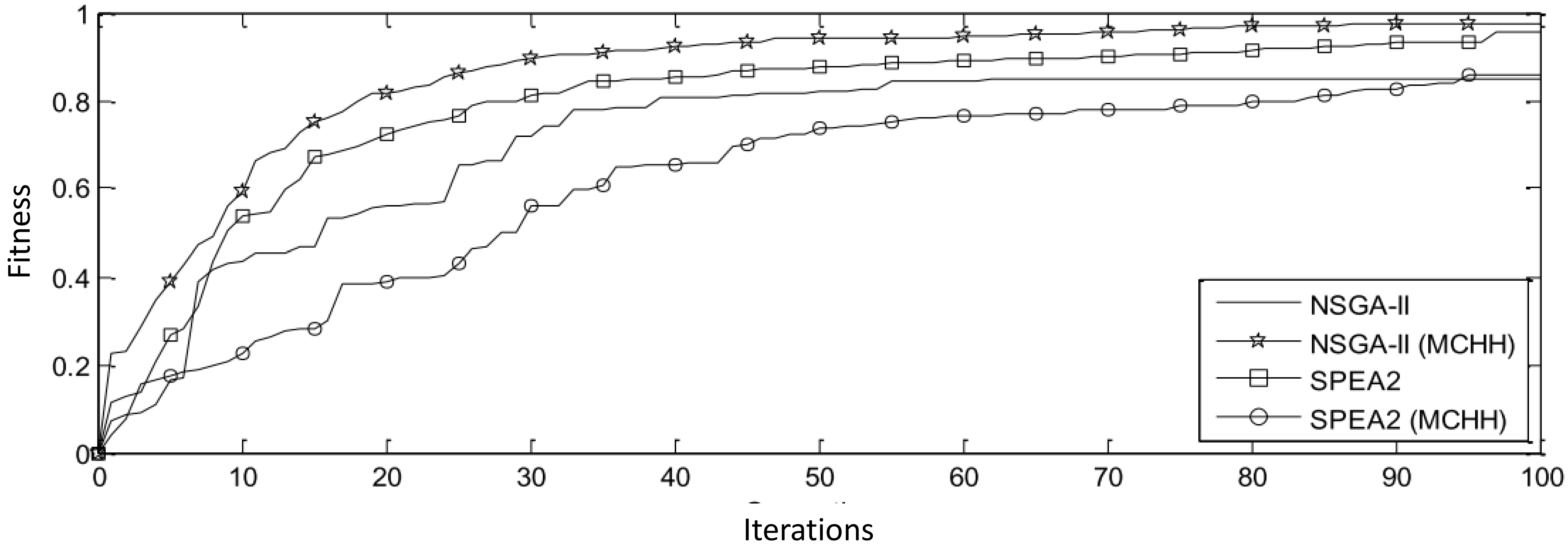
- Fitness of a candidate solution can be estimated using, for example, artificial neural network (for example MOGA-ANN methods)
  - In the beginning all solutions are evaluated completely using a model and ANN is taught
  - When ANN becomes learned enough, the fitness is firts approximated using the ANN
  - If the solution is very good or bad, the actual goodness is simulated using the model, and ANN is trained some more. Otherwise the ANN's estimate is used



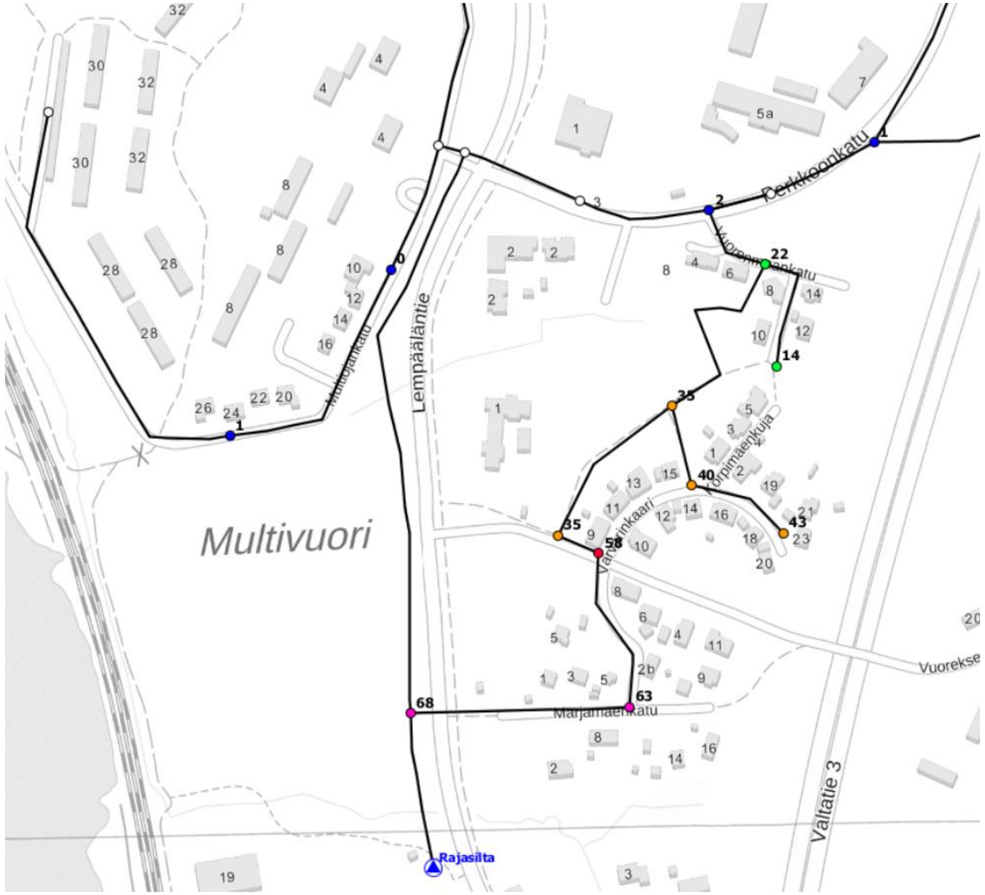




RESULTS GET ITERATIVELY BETTER



LOCATING LEAKS



- DDS + brute force approach for finding a 2000 m³/d leak
- Minimizing the difference between measured and simulated pressures and flows + water tower level
- The higher the value, the more probable location is for the pipe burst
- Not only one answer, but many with different probabilities!